In the Specification

Please amend the specification of this application as follows:

Rewrite paragraph [01] as follows:

--[01] This application claims priority to S.N. 99400551.0, filed in Europe on March 8, 1999 ~~(TI-28234EU)~~; S.N. 98402466.1, filed in Europe on October 6, 1998 ~~(TI-27681EU)~~ and S.N. 98402455.4, filed in Europe on October 6, 1998 ~~(TI-28433EU)~~.--

Rewrite paragraph [26] as follows:

--[26] A description of various architectural features and a description of a complete set of instructions of the microprocessor of Figure 1 is provided in co-assigned ~~application Serial~~ U.S. Patent No. ~~09/410,977 (TI-28433)~~ 6,658,578, which is incorporated herein by reference.--

Rewrite paragraph [75] as follows:

--[75] In one embodiment, the approach is dedicated to write after read (WAR) conflicts as illustrated in Figures 10A and 10B and write after write (WAW) conflicts as illustrated in Figure 9. This technique may be used to reduce stall penalty relative to just a subset of the register file, or to the entire set of registers in the microprocessor. If only a subset of registers is chosen, then the number of shadow register in the stack can be determined by analysis of application code that will be executed on the microprocessor. Likewise, the subset of registers that would benefit ~~form~~ from this stack can be chosen from an application code study defining where the conflicts are and how often they occur.--

Rewrite paragraph [85] as follows:

--[85] Figure 16 is a schematic block diagram of exemplary structure for an interlock control mechanism 1402 using the

arbitration circuitry of Figure 15, for example, for the pipeline 820 of Figure 14. It will be understood that the mechanism could have the same structure for other pipelines, such as the pipeline 850 of Figure ~~30~~ 27. It will be noted that no memory elements (read/write queue) are provided for stall management as the instruction pipeline itself is used to achieve this. For example, a write after write conflict from stage P3 to stage P6 of the pipeline between two consecutive instructions should generate a 3 cycle stall (at stage P3). In practice, the interlock logic generates 3 x 1 cycle consecutive stalls (at stage P3).--